CLIENT	**TrustInSoft**
PROJECT	**White paper on low-level software and firmware security verification**
OBJECTIVE	**Explain the benefits of using exhaustive static analysis to verify low-level code in embedded systems to ensure cybersecurity.**

COPY EXCERPT

From Bare Metal to Kernel Code:

How Exhaustive Static Analysis Can Guarantee Airtight Security in Low-level Software and Firmware

Call or write CopyEngineer to receive a PDF of the complete case study.

Or view/download it online at: bit.ly/TrustInSoft-WP-Low-levelSW-FW

Embedded systems (and businesses) at high risk

According to the consulting firm RSK Cyber Security, embedded systems are particularly prone to cyberattacks.[i]

For businesses, this can present a serious risk, as these devices are directly interconnected with the core network of the company. A coding error in an embedded device can provide an avenue for an attack on the enterprise as a whole. The flaw not only compromises the device; it could take down the company's entire network.

There are several reasons embedded systems are so susceptible. First, they can be attacked through vulnerabilities on two fronts, through both the hardware and the code (software and firmware). Second, integration with the IoT (connectivity) increases the number of attack vectors.

"Another reason is stuffing a small embedded system with many functionalities leads to a lack of security by design," says Praveen Joshi of RSK Cyber Security.[ii]

To save memory space and limit power consumption, developers of these embedded applications often resort to non-standard coding structures While done out of necessity, this practice results in optimized code that makes bugs hard to find.

Common vulnerabilities in low-level code

Many attacks on embedded systems target vulnerabilities caused by bugs in low-level code.

One of the most common types of attacks against embedded firmware and software targets a coding error vulnerability known as memory buffer overflow. This software weakness was ranked #1 on the CWE Top 25 2019 list.[iii] It typically ranks highly from year to year and is most prevalent in the C and C++ programming languages.

"In this type of attack, hackers exploit the system vulnerabilities to swamp the device's memory," says Joshi. "Attackers manually fill the memory buffer allocated to contain the moving data inside the embedded systems. The OS of the embedded system will attempt to record some data in the memory section next to the buffer. But, eventually, it will fail." [iv]

---

[i] Joshi, Praveen, *Common Attacks On Embedded Systems And How To Prevent Them*, RSK Cyber Security, August 2022.

[ii] Ibid.

[iii] *2019 CWE Top 25 Most Dangerous Software Errors*, Mitre, July 2021.

[iv] Joshi, Praveen, *Common Attacks On Embedded Systems And How To Prevent Them*, RSK Cyber Security, August 2022.